6. Haru

Program Name: Haru.java **Input File: haru.dat**

Haru and his friend are playing a game called Spikebikes. In this game, each player programs their bike with a set of directions. The bikes will then follow these movement instructions, leaving spikes behind them.
Bikes will crash if they run into a spike or another bike. Given each bike's starting location and instruction set, determine the outcome of the game!

Input:
The first line of input will consist of a single integer T (1 <= T <= 10), the number of test cases to follow.
Each test case will begin with a single integer N, denoting the length of the instruction sequences.
The next two lines will be of the format X Y S and will denote the starting location and instruction sequences of players one and two, respectively. X and Y will be integers (0 <= X, Y <= 100) and S will be a string of length N containing characters {U, D, L, R} denoting movement in the {+Y, -Y, -X, +X} directions, respectively.

Coordinates are a standard X-Y system, and locations less than zero and greater than 100 contain spikes and will crash a bike.

No test case will contain a head on collision at a spike location.

Output:
For each test case, report one of the following outcomes:

1) DRAW (if no bike crashes)
2) DOUBLE SPIKE (if both bikes hit a spike at the same time)
3) HEAD ON (if the bikes crash into each other)
4) P1 WIN (if player 2 runs into a spike first)
5) P2 WIN (if player 1 runs into a spike first)

Sample input:

```
5
1
5 6 R
6 6 L
3
3 3 RRR
5 4 DDD
1
0 0 U
1 1 L
2
0 0 UR
1 1 DL
2
0 0 UU
1 1 UU
```

Sample output:

```
HEAD ON
P2 WIN
HEAD ON
DOUBLE SPIKE
DRAW
```